CAR-TR-720
CS-TR-3294

DAAH-0493G0419
June 1994

TRACKING A DYNAMIC SET OF FEATURE POINTS

Yi-Sheng Yao
Rama Chellappa*

Department of Electrical Engineering
Center for Automation Research
*Institute for Advanced Computer Studies
University of Maryland
College Park, MD 20742-3275

CAR-TR-720
CS-TR-3294

DAAH-0493G0419
June 1994

TRACKING A DYNAMIC SET OF FEATURE POINTS

Yi-Sheng Yao
Rama Chellappa*

Department of Electrical Engineering
Center for Automation Research
*Institute for Advanced Computer Studies
University of Maryland
College Park, MD 20742-3275

Abstract

This paper presents a model-based algorithm for tracking feature points over a long sequence of monocular noisy images with the ability to include new feature points detected in successive frames. The trajectory for each feature point is modeled by a simple kinematic motion model. A Probabilistic Data Association Filter is first designed to estimate the motion between two consecutive frames. A matching algorithm then identifies the corresponding point to subpixel accuracy and an Extended Kalman Filter (EKF) is employed to continually track the feature point. An efficient way to dynamically include new feature points from successive frames into a tracking list is also addressed. Tracking results for several image sequences are given.

The support of the Advanced Research Projects Agency (ARPA Order No. A422) and the Army Research Office under Grant DAAH-0493G0419 is gratefully acknowledged, as is the help of Sandy German in preparing this paper.

1 Introduction

Motion estimation has been an important topic in the field of computer vision for more than a decade. Based on matches of a few discrete features such as points and lines over two or three frames, many algorithms for estimating the motion of the camera and the structure of the feature points have been proposed. Although linear algorithms result when two or three frames are used, high sensitivity of the estimates to input errors has been observed [1, 2, 11, 13]. In the meantime, the robustness of approaches that use a sequence of images has attracted the attention of many researchers [6, 17, 18, 19]. The issue of finding feature correspondences over a long sequence of images needs to be addressed in such approaches.

Besides manual tracking algorithms, existing techniques for tracking a set of discrete features over a sequence of images generally fall into two categories: two-frame based and long-sequence based.

(1) Two-frame based approaches: In this category, finding feature correspondences over a sequence of images is broken into successive problems of two-view matching. For example, in [16], Weng, Ahuja and Huang used multiple attributes of each image point such as intensity, edgeness and cornerness which are invariant under rigid motion in the image plane along with a set of constraints to compute a dense displacement field and occlusion areas in two images. Cui, Weng and Cohen [9] then used an intensity-based cross-correlation method to refine the two-view matching results and obtain feature point correspondences over the sequence. In [21], Zheng and Chellappa first apply an image registration technique to compensate for the motion of the camera between two consecutive frames. Feature point correspondence problems are then solved by repeatedly identifying the matching points to subpixel accuracy using the correlation matching method.

(2) Long-sequence based approaches: In this category, smoothness constraints are employed to exploit the temporal information existing in the sequence. For example, assuming that the motion of an object does not change abruptly, Sethi and Jain [15] formulated the correspondence problem as an optimization problem. The trajectories of a set of feature points are obtained by searching for a set of trajectories each of which has maximal smoothness. Blostein and Huang [5] used Multistage Hypothesis Testing (MHT) to detect small moving objects in each image; a feature trajectory is determined by repeatedly detecting the same feature point over the sequence. Chang and Aggarwal [7] assumed a 2-D kinematic motion model

and applied Joint Probabilistic Data Association (JPDA) to track line segments, with the ability to initiate or terminate the trajectory of a line segment. Employing a 3-D kinematic motion model and a Mahalanobis distance based matching criterion, Zhang and Faugeras [20] applied an Extended Kalman Filter (EKF) to track a set of line segments. A fading memory type statistical test was suggested to take into account the occlusion and disappearance of line segments.

In this paper, a long sequence based approach is proposed. Finding the trajectory of a feature point over a sequence of images is formulated as a recursive state estimation and measurement identification problem. A discrete 2-D constant translational and rotational motion model is adopted to describe the motion of every feature point. Using the information in other feature points detected in subsequent images, a Probabilistic Data Association Filter (PDAF) [4] is employed to estimate the motion parameters between two consecutive frames. However, because of the imperfect feature detection algorithm, a local image interpolation technique combined with weighted correlation matching, an image differential method, and interpolation of pixel locations are used to identify the matching point to subpixel accuracy. After the identification of corresponding points, an EKF is applied to refine the estimates of the motion parameters. In addition, to maintain a certain number of feature points on the tracking list, the dynamic inclusion of new feature points extracted from successive frames is also considered. We have tested the feature tracking algorithm on several real image sequences commonly employed in motion estimation. Due to space limitations, we present results only on four of these sequences.

The organization of the paper is as follows. Section 2 presents an algorithm for tracking a dynamic set of feature points. The tracking results for four real image sequences are shown in Section 3. Conclusions are presented in Section 4.

2 Feature Point Tracking

In this section, an algorithm for tracking a dynamic set of feature points is presented. The motion model for a feature point moving over a sequence is first formulated. A scheme for estimating the motion between two consecutive frames and procedures for identifying the matching points are then suggested. The issue of the inclusion of new feature points is addressed afterwards.

2.1 Motion Model

To model the motion of a feature point over a sequence of images, a coordinate system xyt shown in Figure 1 is first established with the origin coinciding with the center of the first image and the x-y plane parallel to the image plane at each time instant. Then, assuming that the image center of each frame is located on the t-axis, the coordinates of the center of the k^{th} image are $(0, 0, t_k)^T$. The state vector for a feature point at time t_k is therefore defined as follows:

$$\underline{x}(k) = [x(k)\ y(k)\ v_x(k)\ v_y(k)\ \theta(k)]^T \tag{1}$$

where $(x(k), y(k), t_k)^T$ are the coordinates of a feature point in the k^{th} image, $(v_x(k), v_y(k))^T$ is the associated translational velocity along the (x, y) direction, and $\theta(k)$ is the rotation angle from the $(k-1)^{\text{st}}$ image to the k^{th} image. We describe the motion model of a feature point as well as the relationship between the associated state vector and the image plane coordinates in the form of plant and measurement equations in the following.

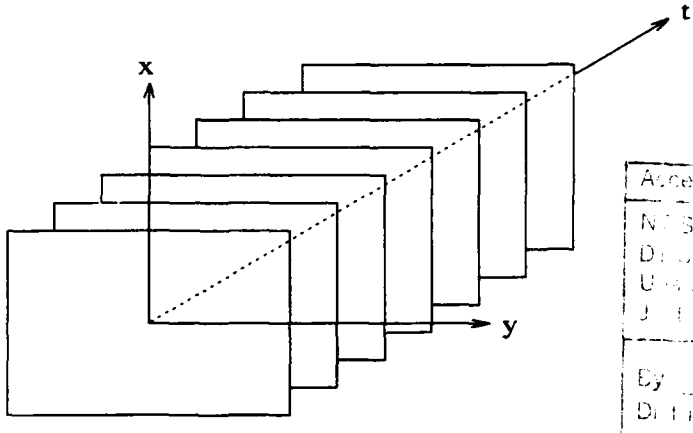


Figure 1: The coordinate system xyt

A. The Plant Equations

Under the assumption that a feature point moves with constant translation and rotation over the sequence, the plant equation for the recursive tracking algorithm can be written as

$$\underline{x}(k+1) = \underline{f}[\underline{x}(k)] + \underline{w}(k+1) \tag{2}$$

where

$$\underline{f}[\underline{x}(k)] = \begin{pmatrix} x(k)\cos\theta(k) - y(k)\sin\theta(k) + v_x(k)T \\ x(k)\sin\theta(k) + y(k)\cos\theta(k) + v_y(k)T \\ v_x(k) \\ v_y(k) \\ \theta(k) \end{pmatrix} \tag{3}$$

and the plant noise $\underline{w}(k)$ is assumed to be zero mean with covariance matrix $Q(k)$. The addition of the plant noise takes into account possible deviations of the true motion from the assumed simple model. The time interval between two consecutive frames is assumed to be T.

B. The Measurement Equations

For each feature point, the measurements used for the corresponding recursive tracking filter at time t_k consist of the image plane coordinates of the corresponding point in the k^{th} image. Thus, the measurement is related to the state vector in (1) by

$$\underline{z}(k) = H\underline{x}(k) + \underline{n}(k) \tag{4}$$

where

$$H = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \end{pmatrix} \tag{5}$$

and the measurement noise $\underline{n}(k)$ is assumed to be zero mean with covariance matrix $R(k)$.

After the plant and measurement equations have been formulated, the EKF can be applied to recursively estimate the motion between two consecutive frames and track the feature point.

2.2 In-frame Motion Estimation

For every feature point being tracked, to simplify the matching algorithm in identifying the corresponding points in successive frames, the motion between two consecutive frames needs to be compensated. In our work, the PDAF which was originally proposed by Bar-Shalom [3] and used for tracking a moving object in a cluttered environment is applied to provide initial estimates of the motion parameters. In the following, assuming that the trajectory of a feature point has been established up to the k^{th} frame of a sequence, procedures for estimating the in-frame motion between the k^{th} and $(k+1)^{\text{st}}$ frames are described. The detailed derivation of the PDAF can be found in [4]; only a brief review is given in the Appendix for the sake of completeness.

First, for a feature point, the predicted location of its corresponding point $\underline{\hat{z}}(k+1|k)$ is obtained as follows:

$$\begin{pmatrix} \cos(\hat{\theta}(k|k)) & -\sin(\hat{\theta}(k|k)) \\ \sin(\hat{\theta}(k|k)) & \cos(\hat{\theta}(k|k)) \end{pmatrix} \begin{pmatrix} \hat{x}(k|k) \\ \hat{y}(k|k) \end{pmatrix} + \begin{pmatrix} \hat{v}_x(k|k) \\ \hat{v}_y(k|k) \end{pmatrix} \tag{6}$$

where $[\hat{x}(k|k), \hat{y}(k|k)), \hat{v}_x(k|k), \hat{v}_y(k|k), \hat{\theta}(k|k)]$ is the estimated state vector at t_k computed by the EKF.

Subsequently, a window centered at $\underline{\hat{z}}(k+1|k)$ is extracted from the $(k+1)^{\text{th}}$ image and the feature point extraction algorithm reported in [14] is applied to the window to identify salient feature points. Since points which are far away from the predicted location are less likely to be correct, a validation gate based on the Mahalanobis distance [8, 10] is constructed to select potential measurements. Specifically, a validation gate centered at $\underline{z}(k+1|k)$ and with parameter γ is defined [4, 8]:

$$V_{k+1}(\gamma) = \left\{ \underline{z} : [\underline{z} - \underline{\hat{z}}(k+1|k)]^T S^{-1}(k+1)[\underline{z} - \underline{\hat{z}}(k+1|k)] \leq \gamma \right\} \tag{7}$$

where $S(k+1)$ is the covariance matrix of the innovation vector $\underline{z} - \underline{\hat{z}}(k+1|k)$, and γ decides the scope of the validation gate and can be obtained from the chi-square distribution table. A set of potential measurements thus consists of the extracted points whose distances are less than γ. The PDAF then combines the information in the potential measurements using (38) in the Appendix to provide estimates of the motion parameters between the k^{th} and $(k+1)^{\text{st}}$ images. For convenience, the resulting estimates are denoted by $\hat{V}_x(k+1|k+1)$, $\hat{V}_y(k+1|k+1)$, and $\hat{\Theta}(k+1|k+1)$ respectively to differentiate them from the outputs from the EKF. For feature points for which none of the extracted points qualify as potential measurements, the predicted motion parameters in (34) are used instead.

2.3 Feature Matching

After the initial estimates of the motion parameters between the k^{th} and $(k+1)^{\text{st}}$ images have been obtained, in order to find corresponding points (or measurements for the EKF), a sequence of steps similar to those in [21] is applied to achieve subpixel accuracy matching. First, a local image registration technique is used to compensate for the motion between two consecutive frames. The resulting compensated image is then compared with the original image and the matching points for the neighboring pixels are identified using weighted correlation matching. However, because of the 3-D motion of the camera, a verification procedure is employed to exclude some possible

wrong matches from the correlation matching before applying the subpixel correction and location interpolation schemes to obtain the corresponding point. In this subsection, the scheme for finding the corresponding point at the $(k+1)^{st}$ frame is described in detail.

A. Window Interpolation

Given two images between which the perspective projection distortion is negligible, the accuracy of using the intensity-based correlation matching method to find the matching point relies on image plane compensation for rotation and scale change. Instead of applying a global registration technique, a local image registration technique which considers small patches of images in which the feature point appears is employed for each feature point.

Since only small patches of two images are considered at any given time, the scale change in the two windows is assumed to be insignificant. The more accurate predicted location of the corresponding point, denoted by $(x'(k+1|k), y'(k+1|k))^T$, can be obtained using the estimates provided by the PDAF as

$$\begin{pmatrix} \cos(\hat{\Theta}(k+1|k+1)) & -\sin(\hat{\Theta}(k+1|k+1)) \\ \sin(\hat{\Theta}(k+1|k+1)) & \cos(\hat{\Theta}(k+1|k+1)) \end{pmatrix} \begin{pmatrix} \hat{x}(k) \\ \hat{y}(k) \end{pmatrix} + \begin{pmatrix} \hat{V}_x(k+1|k+1) \\ \hat{V}_y(k+1|k+1) \end{pmatrix} \tag{8}$$

where $(\hat{x}(k), \hat{y}(k))^T$ is the corresponding point in the k^{th} frame.

Thus, centering on the predicted location and assuming that the pixels near the predicted location undergo the same motion, a window denoted by I_1 is generated using back propagation followed by bilinear interpolation of the intensity function, i.e. for the point whose coordinates are $(x(k+1), y(k+1))^T$ in the $(k+1)^{st}$ frame and which belongs to I_1, the corresponding point in the k^{th} frame is computed by

$$\begin{pmatrix} x(k) \\ y(k) \end{pmatrix} = \begin{pmatrix} \cos(\hat{\Theta}(k+1|k+1)) & \sin(\hat{\Theta}(k+1|k+1)) \\ -\sin(\hat{\Theta}(k+1|k+1)) & \cos(\hat{\Theta}(k+1|k+1)) \end{pmatrix} \begin{pmatrix} x(k+1) - \hat{V}_x(k+1|k+1) \\ y(k+1) - \hat{V}_y(k+1|k+1) \end{pmatrix} \tag{9}$$

Because $(x(k), y(k))^T$ may not be at a grid point, the intensity of the pixel $(x(k+1), y(k+1))^T$ is obtained by interpolating the intensities of the four nearest neighbors of $(x(k), y(k))^T$:

$$g[x(k+1), y(k+1)] = (1-d_x)(1-d_y)g_{11} + d_x(1-d_y)g_{12} + (1-d_x)d_y g_{21} + d_x d_y g_{22} \tag{10}$$

where d_x, d_y are the distances between $(x(k), y(k))^T$ and its neighboring pixel $([x(k)], [y(k)])^T$, and

$\{g_{11}, g_{12}, g_{21}, g_{22}\}$ represent the intensities of the four nearest neighbors of $(x(k), y(k))^T$, i.e.

$$\begin{aligned} d_x &= x(k) - [x(k)] \\ d_y &= y(k) - [y(k)] \end{aligned} \tag{11}$$

and

$$\begin{aligned} g_{11} &= g([x(k)], [y(k)]) \\ g_{12} &= g([x(k)], [y(k)] + 1) \\ g_{21} &= g([x(k)] + 1, [y(k)]) \\ g_{22} &= g([x(k)] + 1, [y(k)] + 1) \end{aligned} \tag{12}$$

Note that $[\bullet]$ in (11) and (12) represents the floor function which converts a real number into an integer.

B. Window Extraction

As in the procedure used in estimating in-frame motion, for each feature point, another window, denoted by I_2, centered at the predicted location of the corresponding point $(x'(k+1|k), y'(k+1|k))^T$ in (8) is extracted from the $(k+1)^{\text{st}}$ image. The correlation matching method described below is then applied to I_1 and I_2 to find the corresponding point.

C. Correlation Matching

Since the motion between the time instants t_k and t_{k+1} has been compensated, a simple intensity-based correlation matching method is employed to find the matching points in I_1 and I_2. Two approaches are possible, as suggested in [21]: a hierarchical matching method (which first uses a large template to achieve coarse matching and then searches for the corresponding point around the neighborhood of the coarse matching result with a small template to achieve better localization) or a weighted correlation matching method. It has been found in our experiments that weighted correlation matching outperforms hierarchical matching not only computationally but also in accuracy. For two points $(x, y)^T \in I_1$ and $(\hat{x}, \hat{y})^T \in I_2$, define the similarity measure as [21]

$$\psi_{g_1 g_2}(x, y; \hat{x}, \hat{y}) = \frac{\sum_{i,j} \gamma_{ij} [g_1(x+i, y+j) - \mu_1][g_2(\hat{x}+i, \hat{y}+j) - \mu_2]}{\sqrt{\sum_{i,j} \gamma_{ij} [g_1(x+i, y+j) - \mu_1]^2} \sqrt{\sum_{i,j} \gamma_{ij} [g_2(\hat{x}+i, \hat{y}+j) - \mu_2]^2}} \tag{13}$$

where

$$\mu_1 = \frac{1}{N} \sum_{i,j} g_1(x+i, y+j) \tag{14}$$

$$\mu_2 = \frac{1}{N}\sum_{i,j} g_2(\hat{x}+i, \hat{y}+j) \tag{15}$$

Here N is the number of pixels in the template Ω, γ_{ij} is the weight associated with points $(x+i, y+j)^T$ and $(\hat{x}+i, \hat{y}+j)^T$, and g_1 and g_2 are the intensity values of the pixels in I_1 and I_2 respectively.

For the point $(x+i, y+j)^T$ or $(\hat{x}+i, \hat{y}+j)^T$ in the template, we define its distance from the center of the template as

$$\max(|i|, |j|)$$

In order to achieve better localization, weights are assigned to pixels based on their distances from the center of the template. In addition, contributions from points of the same distances (i.e. the summation of the corresponding weight coefficients) are restricted to be the same for different levels. Therefore, we choose the weights as follows:

$$\gamma_{ij} = \begin{cases} 1 & (i,j) = (0,0) \\ \frac{c}{8\max(|i|,|j|)} & (i,j) \neq (0,0) \end{cases}$$

where c is a constant and is chosen to account for the relative weights at different levels. Once the weights have been chosen, the matching point for $(x,y)^T$ can be found by searching over a small region in I_2 for the point which has maximal value of the similarity measure (13) with $(x,y)^T$.

It is clear that the above correlation matching method can only match a grid point in I_1 with another grid point in I_2. Since the predicted location of the corresponding point, $(x'(k+1|k), y'(k+1|k))^T$, usually does not coincide with a grid point, the matching points of its four nearest neighbors, $(x_{11}, y_{11})^T, (x_{12}, y_{12})^T, (x_{21}, y_{21})^T$ and $(x_{22}, y_{22})^T$, are first found using the correlation matching method, where

$$\begin{aligned} (x_{11}, y_{11}) &= ([x'(k+1|k)], [y'(k+1|k)]) \\ (x_{12}, y_{12}) &= ([x'(k+1|k)], [y'(k+1|k)]+1) \\ (x_{21}, y_{21}) &= ([x'(k+1|k)]+1, [y'(k+1|k)]) \\ (x_{22}, y_{22}) &= ([x'(k+1|k)]+1, [y'(k+1|k)]+1) \end{aligned} \tag{16}$$

and an interpolation scheme is then applied to obtain the corresponding point.

D. Occlusion and Perspective Distortion Verification

Due to the 3-D motion of the camera, there may exist severe perspective projection distortion between two windows I_1 and I_2, as well as occlusion of feature points. Either case is likely to

introduce large errors when correlation matching is used. A scheme to exclude possible wrong matches for the four neighboring pixels is employed before continuing the tracking process.

For the similarity measure $\psi_{g_1g_2}$ defined in (13), it has been shown that [21]

$$|\psi_{g_1g_2}| \leq 1 \tag{17}$$

and the following equality holds:

$$\psi_{g_1g_2} = \begin{cases} 1, & \text{if } g_1(x+i,y+j)-\mu_1 = g_2(\hat{x}+i,\hat{y}+j)-\mu_2 \\ -1, & \text{if } g_1(x+i,y+j)-\mu_1 = -[g_2(\hat{x}+i,\hat{y}+j)-\mu_2] \end{cases} \quad \forall (i,j) \in \Omega \tag{18}$$

If the perspective projection distortion between two windows is large or if occlusion occurs, it is likely that the similarity measures corresponding to the four neighboring pixels will be small. A threshold, say TH, is set to account for wrong matches. Denote the matches resulting from correlation matching for the four neighboring pixels by $(x_{11}, y_{11}; \hat{x}_{11}, \hat{y}_{11})$, $(x_{12}, y_{12}; \hat{x}_{12}, \hat{y}_{12})$, $(x_{21}, y_{21}; \hat{x}_{21}, \hat{y}_{21})$ and $(x_{22}, y_{22}; \hat{x}_{22}, \hat{y}_{22})$. Three cases are considered in the following.

Case 1: More than two matching pairs have similarity measures less than TH.

In this case, the possibility that the feature point is occluded in I_2 or that severe distortion exists in the two windows is very high. We remove the feature point from the tracking list.

Case 2: Three matching pairs have similarity measures greater than TH.

Because there is only one matching pair with a high possibility of a wrong match, the feature point is assumed to exist and the perspective projection distortion between I_1 and I_2 is considered not to be too severe. An extrapolation scheme is therefore introduced to correct the wrong match. Without loss of generality, let $(x_{22}, y_{22}; \hat{x}_{22}, \hat{y}_{22})$ be the matching pair whose similarity measure is less than TH. Since the four neighbors of the feature point form a square in I_1, the four matching points are assumed to form a parallelogram in I_2. The matching point for $(x_{22}, y_{22})^T$ is then recalculated by

$$\begin{cases} \hat{x}_{22} = \hat{x}_{12} + \hat{x}_{21} - \hat{x}_{11} \\ \hat{y}_{22} = \hat{y}_{12} + \hat{y}_{21} - \hat{y}_{11} \end{cases} \tag{19}$$

and the tracking process continues.

Case 3: All the four matching pairs have similarity measures greater than TH.

In this case, the four similarity measures show high similarity between the two windows. Therefore the outputs from the correlation matching are considered reliable and tracking continues.

E. Subpixel Accuracy Correction

For tracking over a sequence of images, the quantization errors in matching grid points to grid points accumulate, resulting in deviations in the trajectories. In order to reduce accumulated errors, the matches obtained from the correlation matching method need to be refined before applying the location interpolation scheme to find the corresponding points.

Since a good initial match has been obtained, the image differential method [12, 21] provides a simple and effective way to achieve subpixel accuracy matching. Assuming that $(x, y)^T \in I_1$ is matched to $(\hat{x}, \hat{y})^T \in I_2$ and the original intensity function $g_2(\hat{x}, \hat{y})$ is offset by (δ_x, δ_y) relative to the interpolated intensity function $g_1(x, y)$, i.e.

$$g_2(\hat{x}, \hat{y}) = g_1(x - \delta_x, y - \delta_y) \tag{20}$$

Then the difference in $g_1(x, y)$ and $g_2(\hat{x}, \hat{y})$ can be written as [12, 21]

$$\begin{aligned} d(x, y; \hat{x}, \hat{y}) &= g_1(x, y) - g_2(\hat{x}, \hat{y}) \\ &= g_1(x, y) - g_1(x - \delta_x, y - \delta_y) \\ &\approx \frac{\partial g_1(x, y)}{\partial x}\delta_x + \frac{\partial g_1(x, y)}{\partial y}\delta_y \end{aligned} \tag{21}$$

For each matching pair $(x, y; \hat{x}, \hat{y})$, suppose that the above approximation holds for a small neighborhood of size $(2\omega_d + 1) \times (2\omega_d + 1)$; then a set of equations can be formed as follows:

$$\vec{D} = G\vec{\Delta} \tag{22}$$

where

$$\vec{D} = \begin{pmatrix} d(x - w_d, y - w_d; \hat{x} - w_d, \hat{y} - w_d) \\ \vdots \\ d(x, y; \hat{x}, \hat{y}) \\ \vdots \\ d(x + w_d, y + w_d; \hat{x} + w_d, \hat{y} + w_d) \end{pmatrix} \tag{23}$$

$$G = \begin{pmatrix} \frac{\partial g_1(x-w_d, y-w_d)}{\partial x} & \frac{\partial g_1(x-w_d, y-w_d)}{\partial y} \\ \vdots & \vdots \\ \frac{\partial g_1(x,y)}{\partial x} & \frac{\partial g_1(x,y)}{\partial y} \\ \vdots & \vdots \\ \frac{\partial g_1(x+w_d, y+w_d)}{\partial x} & \frac{\partial g_1(x+w_d, y+w_d)}{\partial y} \end{pmatrix} \tag{24}$$

and

$$\vec{\Delta} = \begin{pmatrix} \delta_x \\ \delta_y \end{pmatrix} \tag{25}$$

The offset vector $\vec{\Delta}$ is then the least square solution of (22) and can be obtained as

$$(G^T G)^{-1} G^T \vec{D} \tag{26}$$

Thus, $(x, y)^T$ is matched to $(\hat{x} + \delta_x, \hat{y} + \delta_y)^T$ which achieves subpixel accuracy matching. In our experiments, a neighborhood with $\omega_d = 3$ was employed.

F. Location Interpolation

After the matching points of the four nearest neighbors have been found to subpixel accuracy, the point corresponding to the feature point is obtained by the location interpolation scheme described below [21].

For each matching pair $(x, y; \hat{x}, \hat{y})$, assume that the relationship between $(x, y)^T$ and $(\hat{x}, \hat{y})^T$ can be expressed as

$$\begin{cases} \hat{x} &= \hat{\alpha}_1 x + \hat{\alpha}_2 y + \hat{\alpha}_3 xy + \hat{\alpha}_4 \\ \hat{y} &= \hat{\beta}_1 x + \hat{\beta}_2 y + \hat{\beta}_3 xy + \hat{\beta}_4 \end{cases} \tag{27}$$

or expressed relative to the match of $(x_{11}, y_{11})^T$ as

$$\begin{cases} \hat{x} - \hat{x}_{11} &= \alpha_1 (x - x_{11}) + \alpha_2 (y - y_{11}) + \alpha_3 (x - x_{11})(y - y_{11}) + \alpha_4 \\ \hat{y} - \hat{y}_{11} &= \beta_1 (x - x_{11}) + \beta_2 (y - y_{11}) + \beta_3 (x - x_{11})(y - y_{11}) + \beta_4 \end{cases} \tag{28}$$

Then from the four matching pairs and the relationships between the four nearest neighbors in (16), the coefficients α_i and β_i, $i = 1, \ldots, 4$ are [21]

$$\begin{cases} \alpha_1 &= \hat{x}_{21} - \hat{x}_{11} \\ \beta_1 &= \hat{y}_{21} - \hat{y}_{11} \end{cases} \tag{29}$$

$$\begin{cases} \alpha_2 &= \hat{x}_{12} - \hat{x}_{11} \\ \beta_2 &= \hat{y}_{12} - \hat{y}_{11} \end{cases} \tag{30}$$

$$\begin{cases} \alpha_3 &= \hat{x}_{22} + \hat{x}_{11} - \hat{x}_{12} - \hat{x}_{21} \\ \beta_3 &= \hat{y}_{22} + \hat{y}_{11} - \hat{y}_{12} - \hat{y}_{21} \end{cases} \tag{31}$$

and

$$\begin{cases} \alpha_4 &= 0 \\ \beta_4 &= 0 \end{cases} \tag{32}$$

Substituting (29–32) into (28), the feature point location interpolation formula can be written as

$$\begin{cases} \hat{x} &= \hat{x}_{11} + (\hat{x}_{21} - \hat{x}_{11})\epsilon_x + (\hat{x}_{12} - \hat{x}_{11})\epsilon_y + (\hat{x}_{22} + \hat{x}_{11} - \hat{x}_{12} - x_{21})\epsilon_x\epsilon_y \\ \hat{y} &= \hat{y}_{11} + (\hat{y}_{21} - \hat{y}_{11})\epsilon_x + (\hat{y}_{12} - \hat{y}_{11})\epsilon_y + (\hat{y}_{22} + \hat{y}_{11} - \hat{y}_{12} - \hat{y}_{21})\epsilon_x\epsilon_y \\ \epsilon_x &= x - x_{11} \\ \epsilon_y &= y - y_{11} \end{cases} \tag{33}$$

It has been shown in [21] that if the quadrangle formed by $(\hat{x}_{11}, \hat{y}_{11})^T, (\hat{x}_{12}, \hat{y}_{12})^T, (\hat{x}_{21}, \hat{y}_{21})^T$ and $(\hat{x}_{22}, \hat{y}_{22})^T$ is convex, the corresponding point interpolated from (33) is also inside the quadrangle.

This completes the task of identifying the point corresponding to a feature point in the $(k+1)^{st}$ image. The EKF can now use this matching point as the measurement to update the corresponding state vector as well as the covariance matrix, and the algorithm is ready to continue tracking the feature point to the next time instant.

2.4 Inclusion of New Features

When tracking a feature point over a long sequence, it is possible that the point moves out of the field of view or is occluded by the other objects after some time instant. This results in a decrease in the number of feature points being tracked. In addition, because of the motion of the camera, features not detected at earlier instants are likely to be identified later. It is therefore necessary to consider a strategy for including new feature points extracted from the successive frames. In our work, an extracted point is considered to be a new feature point if it does not correspond to any point currently being tracked. Furthermore, instead of initiating tracks for all new feature points, which results in a rapid growth in the number of feature points, validation gates as defined in (7) are employed to screen the newly detected feature points. A new feature point is added to the tracking list if it lies outside all the validation gates associated with the feature points currently being tracked.

For example, in Figure 2, eight validation gates which correspond to the eight feature points in the tracking list are formed. Another nine feature points extracted from the current frame are also shown. Since only z_1, z_6 and z_9 do not fall into any validation gate, they are added to the tracking list.

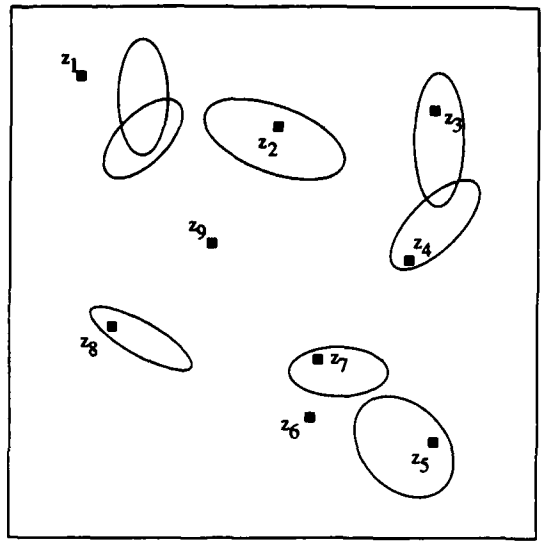


Figure 2: The inclusion of new feature points: eight validation gates and nine newly extracted feature points

In other words, the newly extracted feature points are not tracked if they are too close to the currently tracked feature points. This is particularly useful for estimating the motion of the camera since uniformly distributed points are more likely to cancel out the effects of imperfect knowledge of the camera parameters such as the imaging center and the field of view. The proposed scheme not only takes into account the decrease in the number of feature points on the tracking list but also prevents the number of feature points from growing too fast since as the number of feature points on the tracking list increases, the image region covered by the validation gates also grows.

3 Experimental Results

In this section, tracking results are presented for four real image sequences taken by cameras undergoing different types of motion. For each sequence, in addition to the trajectory termination criterion in Section 2.3, depending on the size of the area correlation mask, feature points too close to the image boundary are removed. A tracking list which contains the matching points as well as the new feature points is created and updated at every frame. For visual purposes, only the trajectories of the feature points tracked from the first frame as well as the new feature points added to the tracking list at subsequent time instants are displayed. The dynamic behavior of the algorithm is shown in a table which lists the number of feature point trajectories being maintained or removed from the tracking list and the number of new points selected from every frame.

3.1 UMASS PUMA2 Sequence

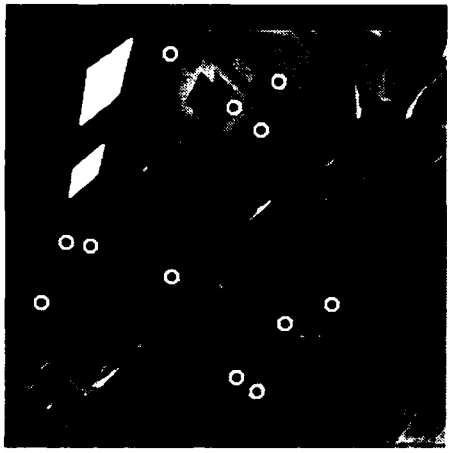
The first sequence is known as the UMASS PUMA2 sequence; it consists of thirty 256×256 frames. The camera is connected to the end of a PUMA robot arm and rotates about a rotation center which is close to the image center. The tracking results for a set of manually selected points which was used in [18] to estimate the motion of the camera are shown in Figure 3. Figure 4 shows the trajectories for a set of feature points automatically extracted from the first frame by the algorithm reported in [14]; the trajectories are shown up to the 7^{th}, 13^{th}, 19^{th}, 25^{th} and 30^{th} frames. The motion parameters corresponding to the two points marked in Figure 4(a) computed by the EKF are displayed in Figure 5. Note that the coordinate system illustrated in Figure 1 has been changed, with the x-axis pointing downward instead of upward for convenience. Since the rotation center does not coincide with the image center, a nonzero translational velocity is observed for both points. The number of feature points being tracked varies with time, as shown in Table 1. The new feature points extracted by the feature extraction algorithm from frames 3, 7, 13, 19, 25 and 30 are shown in Figure 6, in addition to the labeled points which were added to the tracking list at different time instants. As seen in Table 1, the algorithm for adding new points to the tracking list efficiently maintains the number of points on the list.

Table 1: The number of feature points in the tracking list for the UMASS PUMA2 Sequence

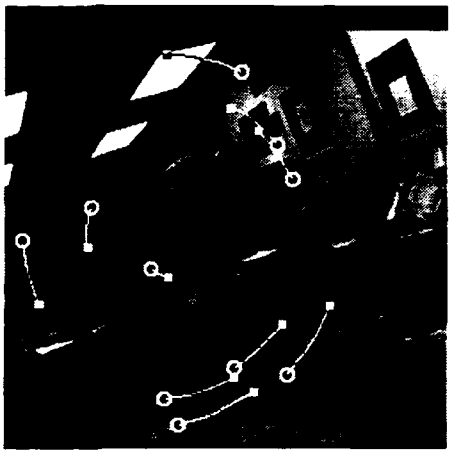
frame number	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
# of points in the list	0	21	19	28	31	35	38	40	41	40	41	43	43	44	46
# of points extracted	23	23	22	26	24	27	29	29	28	28	25	16	19	20	24
# of new points	23	0	9	4	5	5	3	3	4	4	3	1	2	2	5
frame number	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30
# of points in the list	49	48	49	48	50	49	51	50	53	50	52	52	53	54	53
# of points extracted	18	21	20	21	20	21	20	19	23	25	24	24	27	24	15
# of new points	2	3	1	3	1	3	0	4	0	3	0	3	2	1	1

3.2 Coke Can Sequence

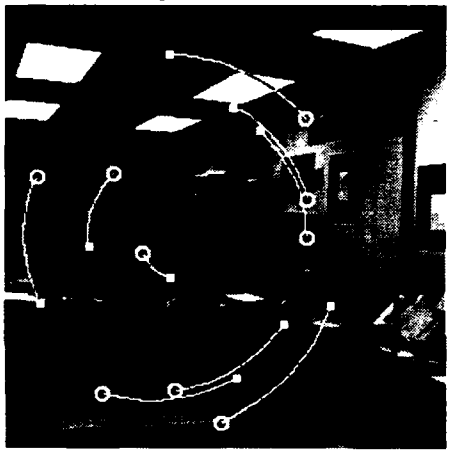
The second sequence is the Coke Can Sequence, in which the camera is approaching the scene, with the Focus of Expansion (FOE) located on the coke can. Fifteen frames chosen from the densely sampled sequence, spaced 10 frames apart, are used. The original 512×512 images are down-sampled to 256×256 before applying the algorithm. The resulting trajectories from the first frame to the 5^{th}, 10^{th} and 15^{th} frames are shown in Figure 7. The estimated motion parameters of the



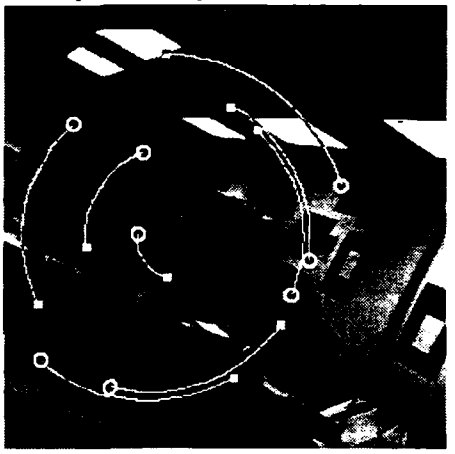
(a) Feature points in the first frame



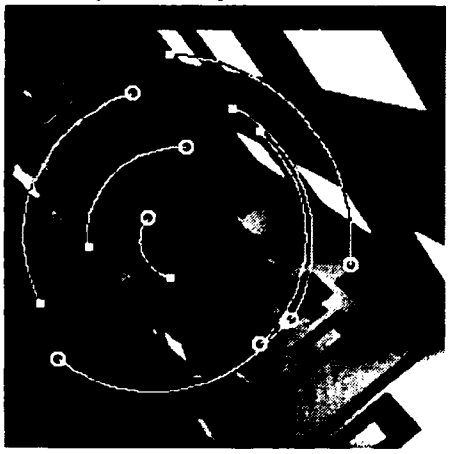
(b) Trajectories up to the seventh frame



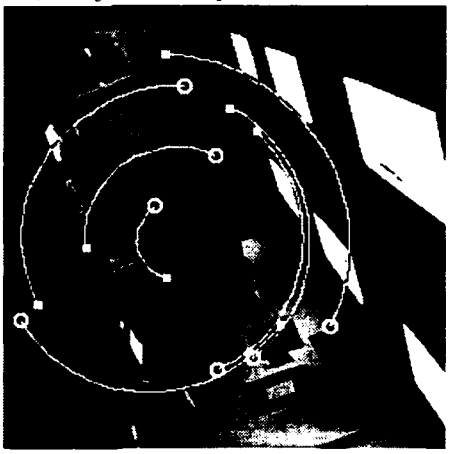
(c) Trajectories up to the 13^{th} frame



(d) Trajectories up to the 19^{th} frame

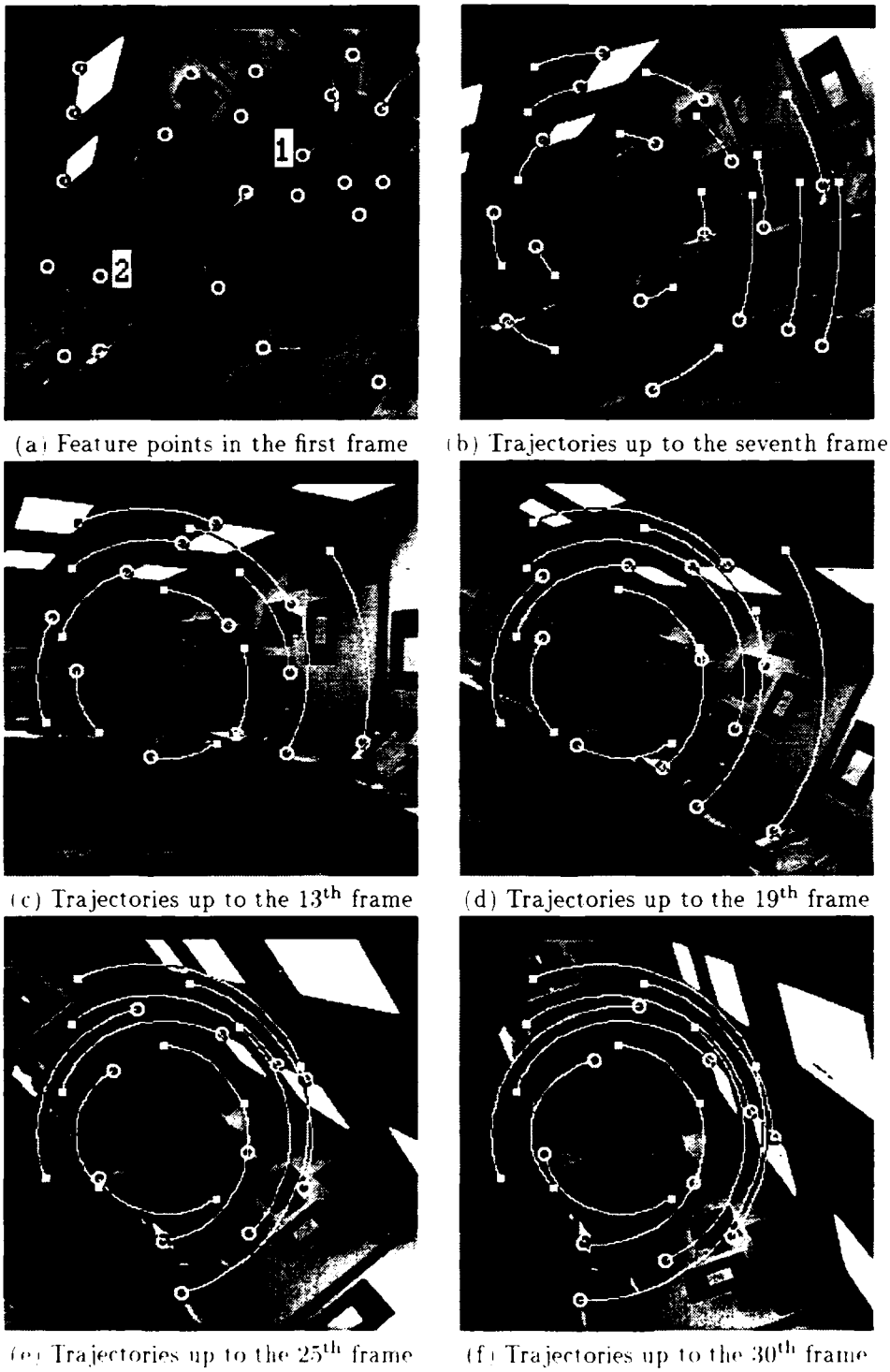


(e) Trajectories up to the 25^{th} frame



(f) Trajectories up to the 30^{th} frame

Figure 3: Trajectories for the UMASS PUMA2 Sequence (manually selected points)



(a) Feature points in the first frame (b) Trajectories up to the seventh frame

(c) Trajectories up to the 13^{th} frame (d) Trajectories up to the 19^{th} frame

(e) Trajectories up to the 25^{th} frame (f) Trajectories up to the 30^{th} frame

Figure 4: Trajectories for the UMASS PUMA2 Sequence (automatically extracted points)

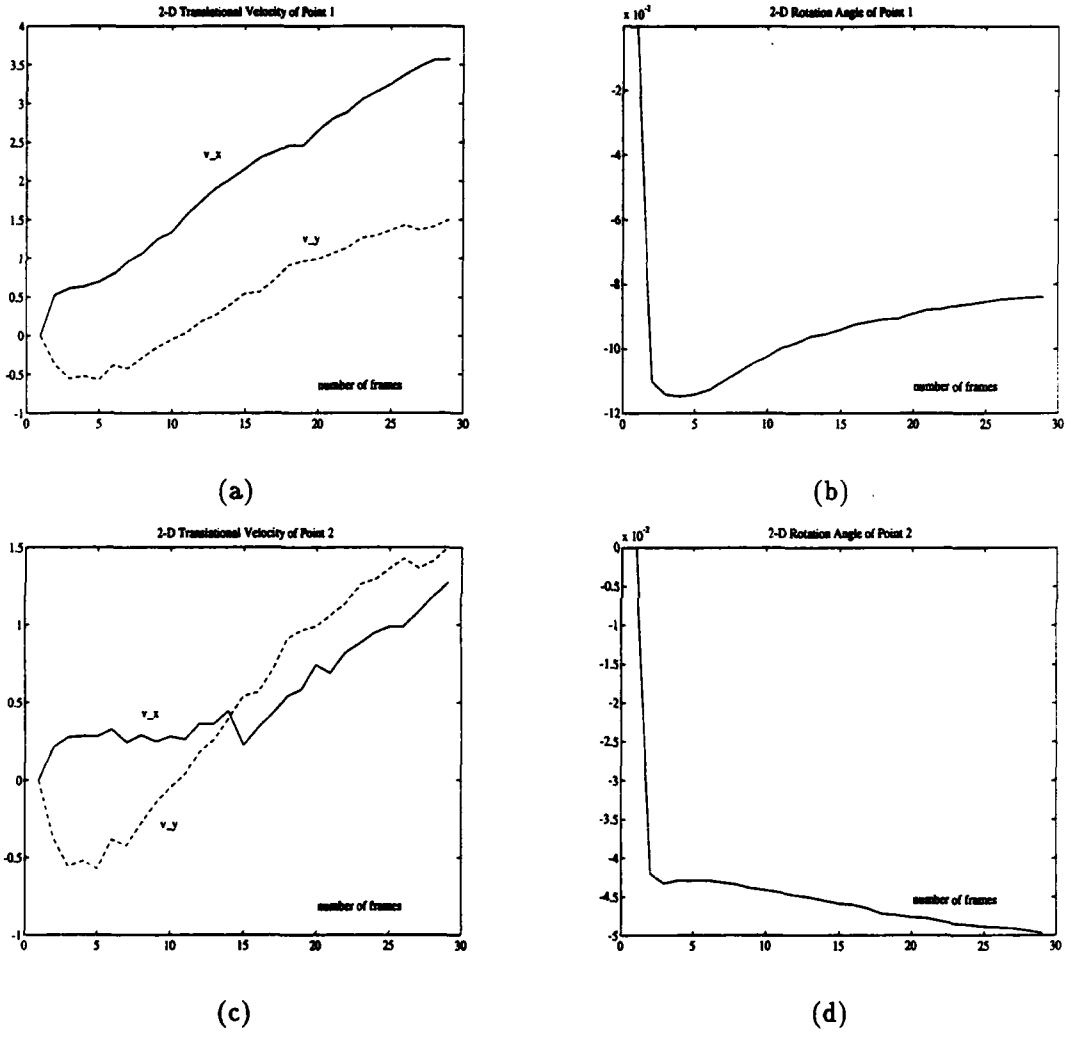
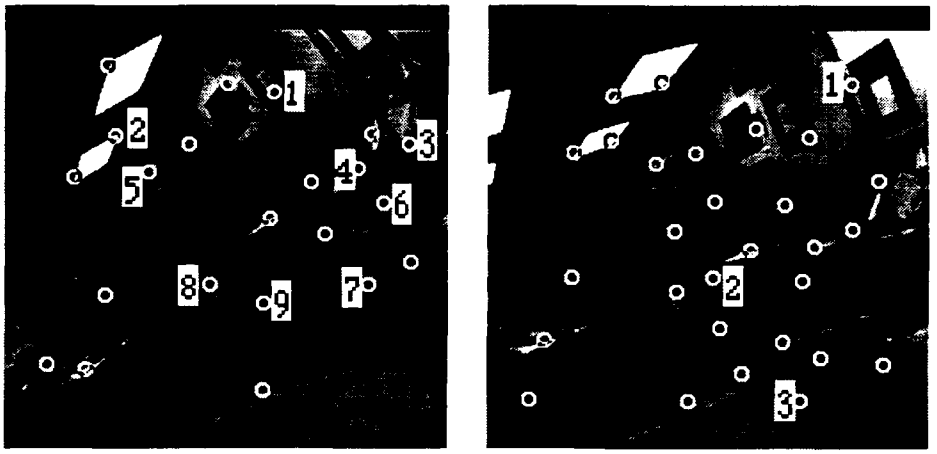
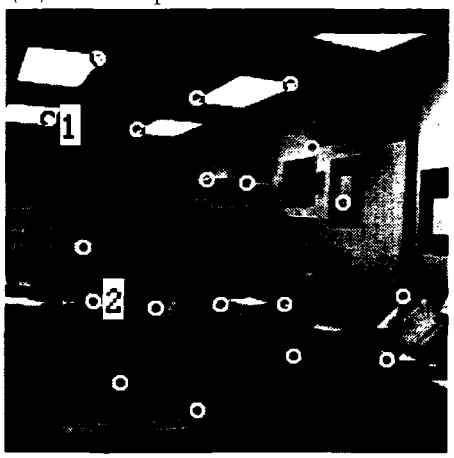


Figure 5: Motion parameters for the UMASS PUMA2 Sequence computed by the Kalman Filter

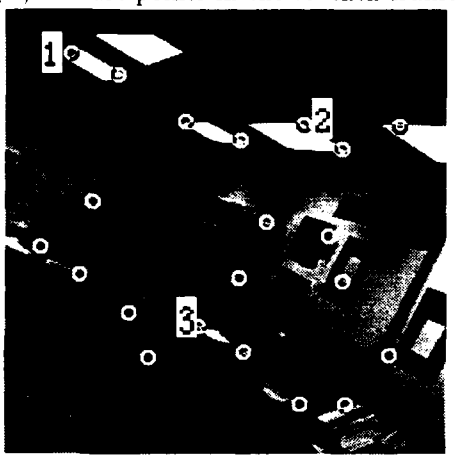


(a) Feature points in the third frame

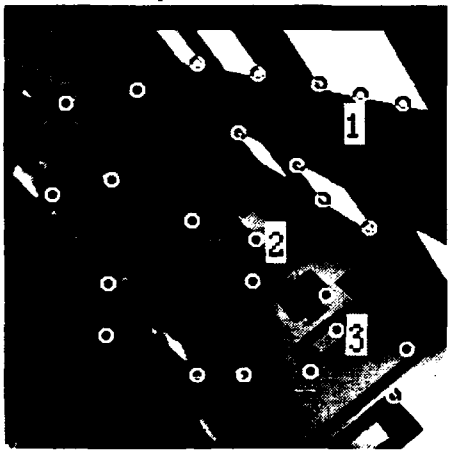
(b) Feature points in the seventh frame



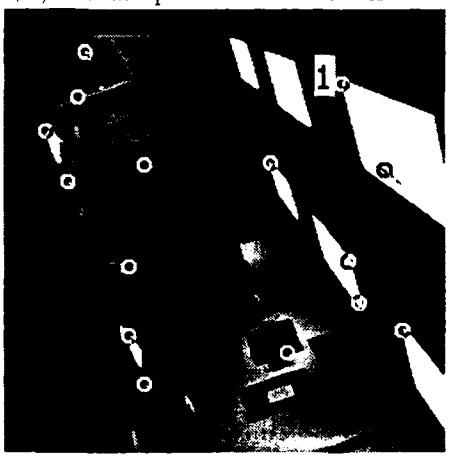
(c) Feature points in the 13^{th} frame



(d) Feature points in the 19^{th} frame



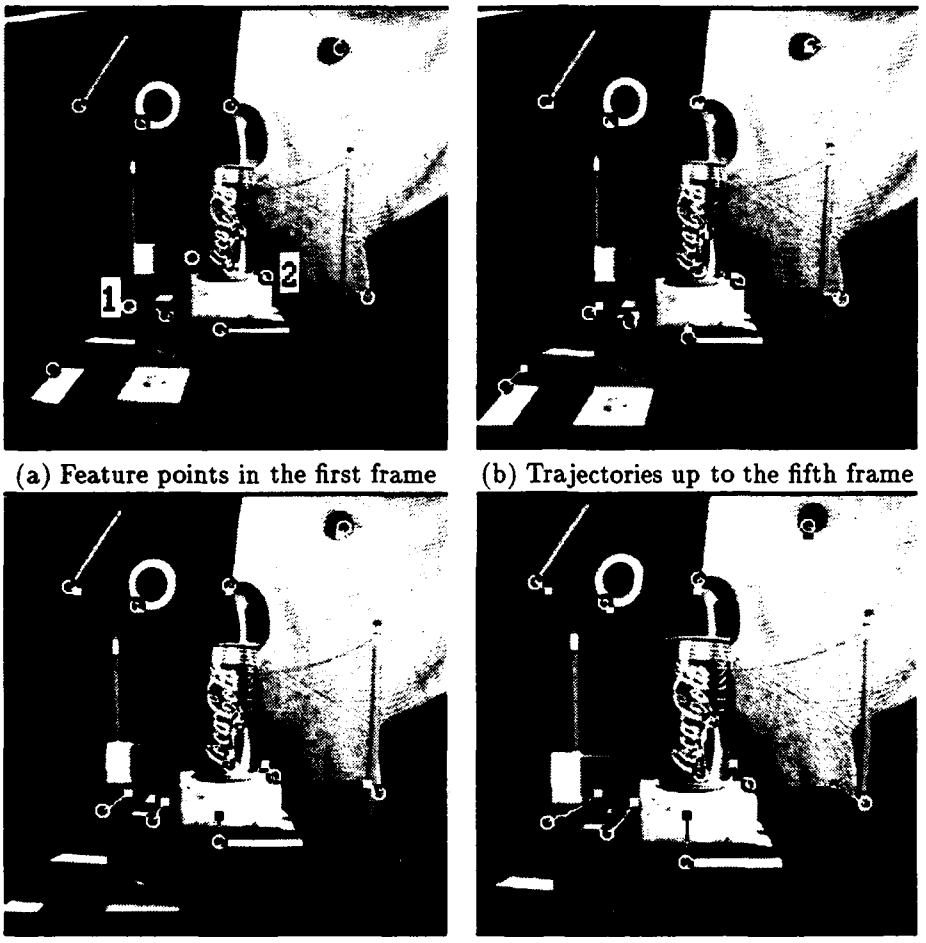
(e) Feature points in the 25^{th} frame



(f) Feature points in the 30^{th} frame

Figure 6: Automatically detected new feature points in the UMASS PUMA2 Sequence

two points marked in Figure 7(a) are displayed in Figure 8. As seen from the figures, because of the pure translation of the camera, the trajectories of the feature points diverge from the FOE and are well described by the motion model. Table 2 lists the number of tracked feature points at each time instant. The new feature points added at the 2^{nd}, 5^{th}, 10^{th} and 15^{th} frames are marked in Figure 9.



(a) Feature points in the first frame (b) Trajectories up to the fifth frame

(c) Trajectories up to the tenth frame (d) Trajectories up to the 15^{th} frame

Figure 7: Trajectories for the Coke Can Sequence

3.3 Rocket ALV Sequence

The third sequence is the 30-frame UMASS Rocket ALV Sequence. Again, the 512×512 images are down-sampled to 256×256 before applying the algorithm. In this sequence, the camera is mounted

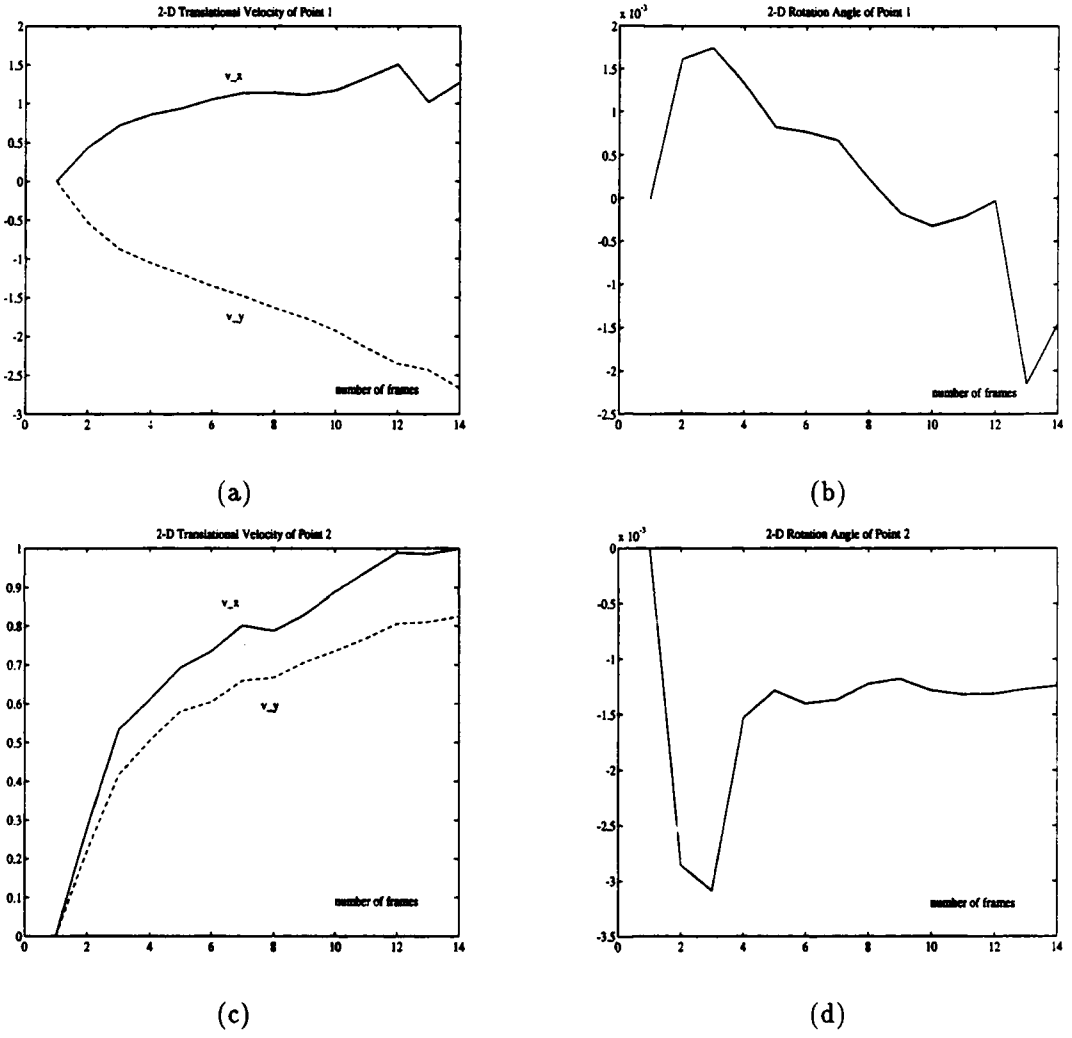
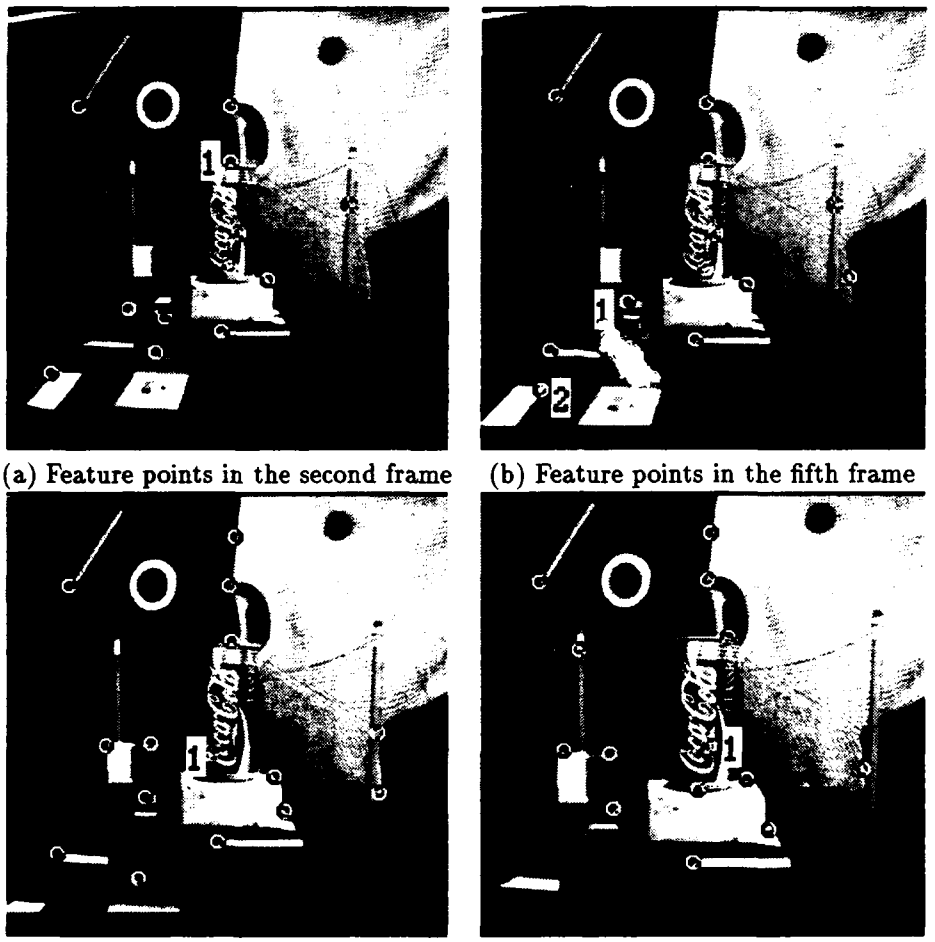


Figure 8: Motion parameters for the Coke Can Sequence computed by the Kalman Filter



(a) Feature points in the second frame (b) Feature points in the fifth frame

(c) Feature points in the tenth frame (d) Feature points in the 15^{th} frame

Figure 9: Automatically detected new feature points in the Coke Can Sequence

Table 2: The number of feature points in the tracking list for the Coke Can Sequence

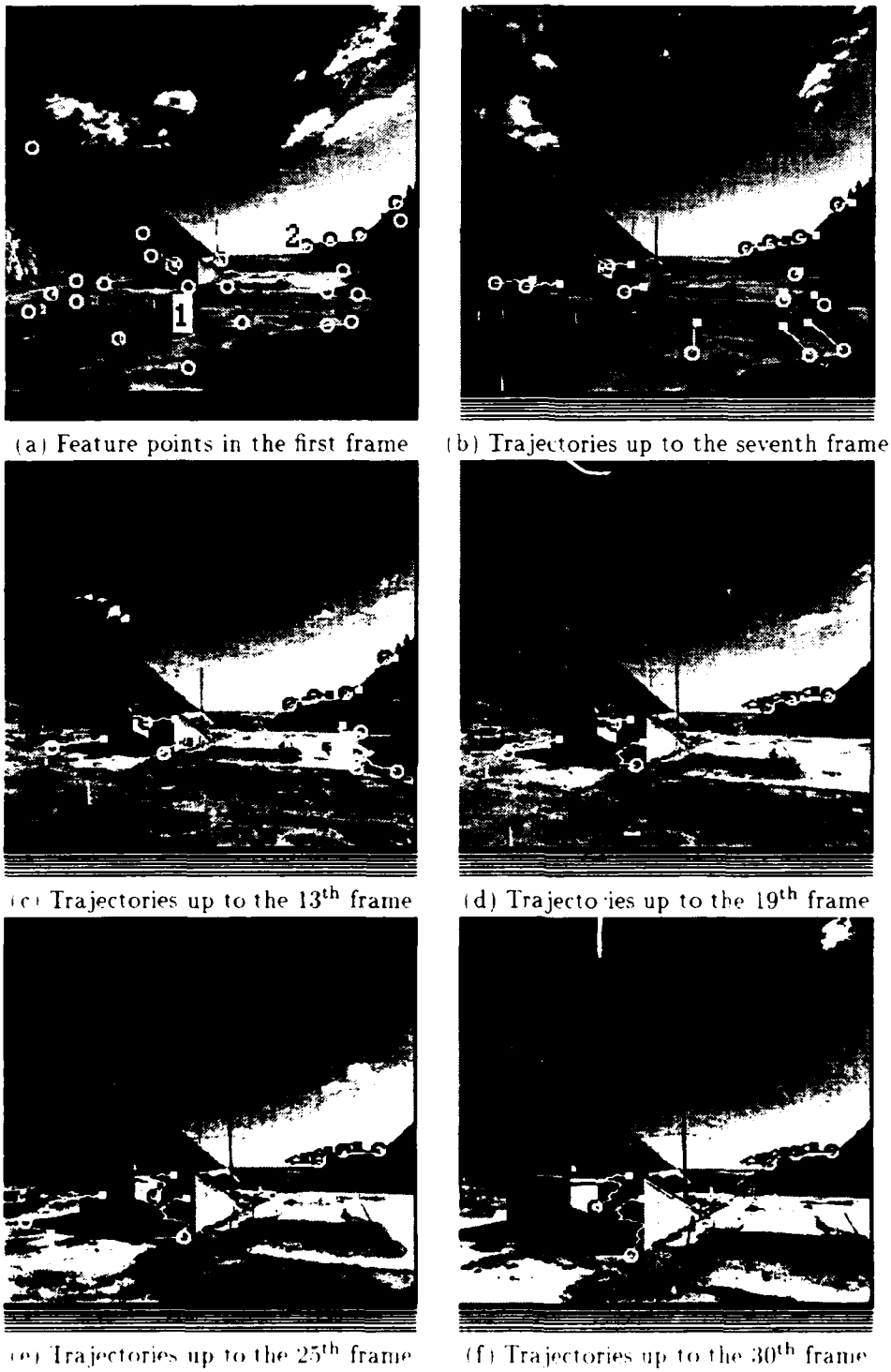
frame number	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
# of points in the list	0	12	13	16	18	20	22	22	22	26	27	28	28	28	28
# of points extracted	13	12	15	12	13	17	15	15	16	15	14	17	15	17	14
# of new points	13	1	4	2	2	2	1	1	5	1	1	2	0	1	1

on the vehicle which appears to be moving along a straight line to the left and into the image plane with almost no rotation. Due to the uneven terrain, the motion of the camera is not smooth. The trajectories for the feature points up to the 7^{th}, 13^{th}, 19^{th}, 25^{th} and 30^{th} frames are shown in Figure 10. Figure 11 displays the motion trajectories corresponding to the two points marked in Figure 10(a). The uneven motion of the camera results in the motion trajectories in Figure 11 being more jerky than those in Figure 5 and Figure 8. Table 3 lists the number of feature points on the tracking list.[1] The extracted feature points as well as the new points selected by the proposition in Section 2.4 from the 2^{nd}, 7^{th}, 13^{th}, 19^{th}, 25^{th} and 30^{th} frames are shown in Figure 12. As seen from the figures, many feature points move out of the field of view in the first few frames. It is therefore necessary to include new feature points when they become available. Also, it is apparent from the sequence that the vehicle has an abrupt change in heading direction at the 16^{th} and 20^{th} frames, but the algorithm still keeps tracking most of the feature points.

Table 3: The number of feature points in the tracking list for the UMASS Rocket ALV Sequence

frame number	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
# of points in the list	0	19	19	23	22	22	19	16	19	21	24	22	22	23	21
# of points extracted	25	20	16	12	14	13	16	16	13	18	14	21	16	17	14
# of new points	25	4	6	1	3	0	7	8	6	6	2	2	3	1	1
frame number	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30
# of points in the list	19	19	18	18	21	19	24	25	27	29	28	26	27	25	25
# of points extracted	19	19	19	18	22	15	17	16	18	18	17	16	19	19	18
# of new points	2	1	3	5	1	7	1	4	2	3	3	2	4	5	2

[1]The feature points on the clouds are manually removed from the tracking list because of their nonrigid shapes, and so are the feature points detected at the bottom of each image due to the strip patterns.



(a) Feature points in the first frame

(b) Trajectories up to the seventh frame

(c) Trajectories up to the 13th frame

(d) Trajectories up to the 19th frame

(e) Trajectories up to the 25th frame

(f) Trajectories up to the 30th frame

Figure 10: Trajectories for the Rocket ATV Sequence

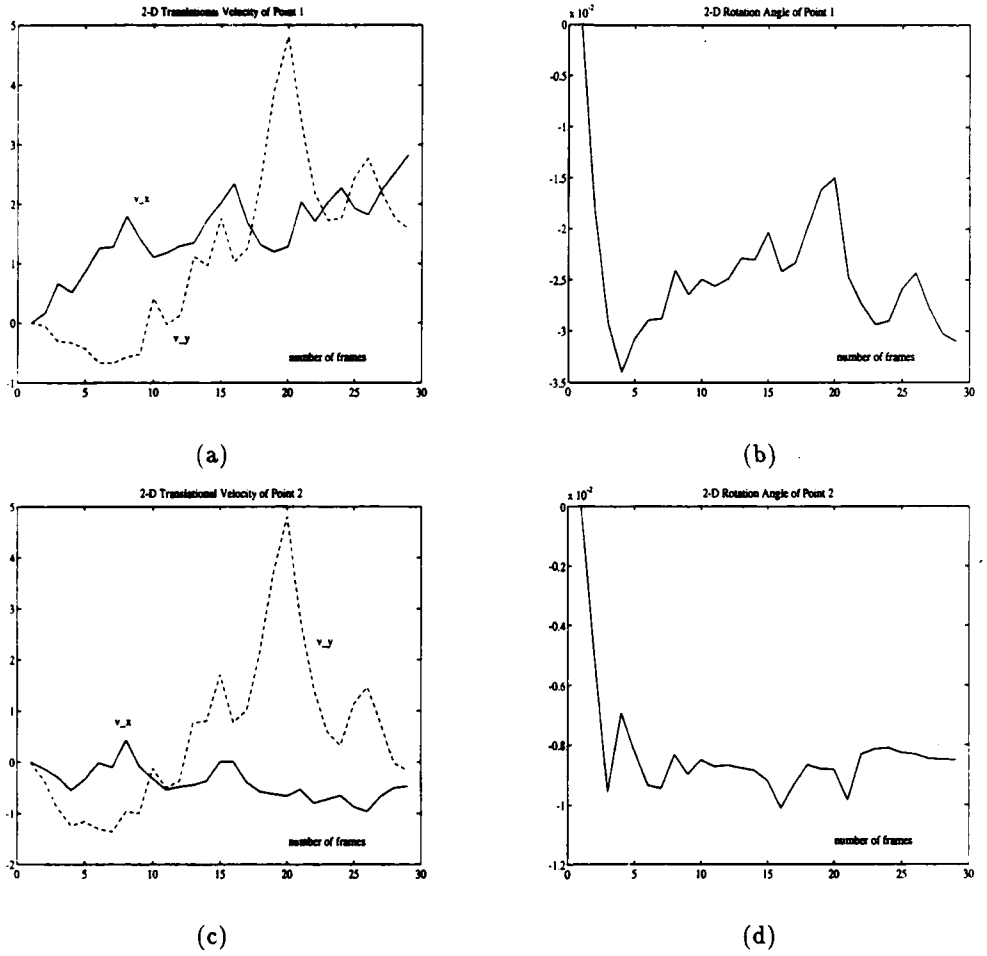
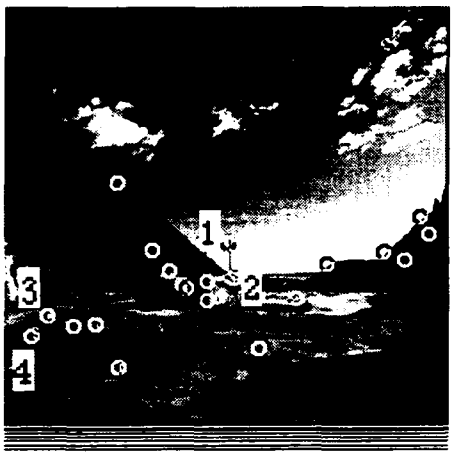
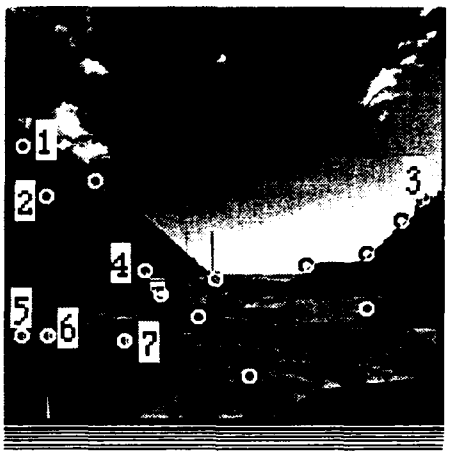


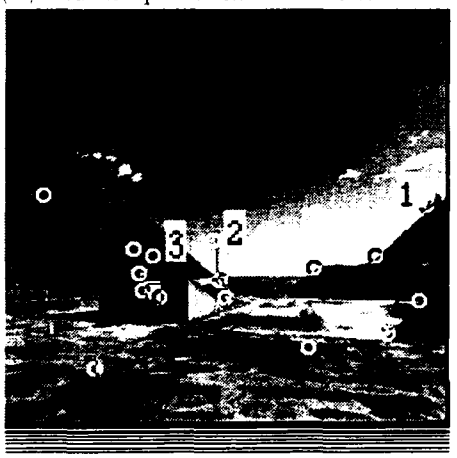
Figure 11: Motion parameters for the Rocket ALV Sequence computed by the Kalman Filter



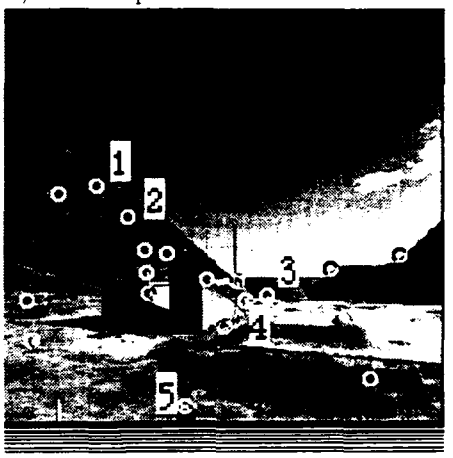
(a) Feature points in the second frame



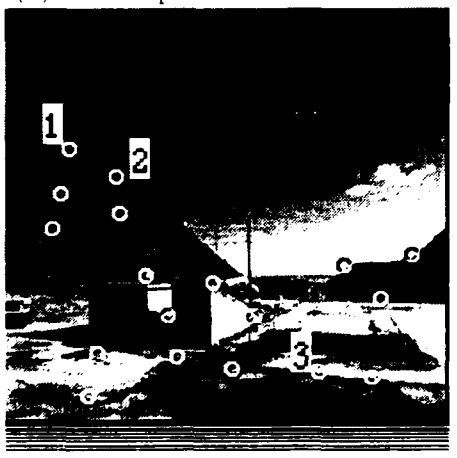
(b) Feature points in the seventh frame



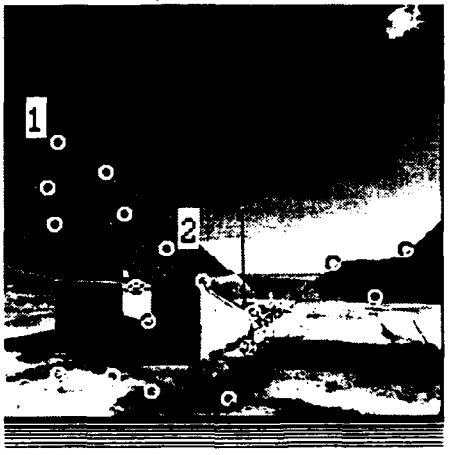
(c) Feature points in the 13^{th} frame



(d) Feature points in the 19^{th} frame



(e) Feature points in the 25^{th} frame



(f) Feature points in the 30^{th} frame

Figure 12: Automatically detected new feature points in the Rocket ALV Sequence

3.4 Martin Marietta R3 Sequence

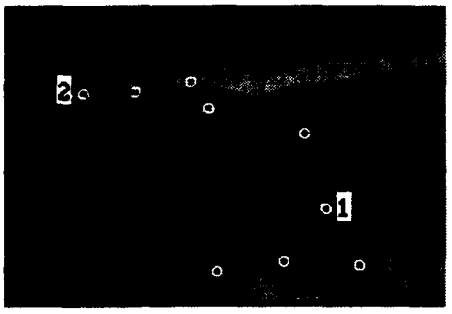
The last sequence is one of the four sequences distributed by Martin Marietta. As in the third sequence, the camera is mounted on a vehicle and the images are taken when the vehicle is moving through an outdoor environment. The original sequence consists of densely sampled images of size 347 $\times$ 238; only thirty frames, five frames apart, were used in the experiment. During the acquisition of the images, the vehicle moves to the right and slightly into the scene. Figure 13 shows the trajectories of a set of feature points from the first frame to the 7^{th}, 13^{th}, 19^{th}, 25^{th} and 30^{th} frames. As seen from the figures, the points on the mountain are far away from the vehicle resulting in small movements on the image plane. The computed motion parameters of the two points marked in Figure 13 are shown in Figure 14. The nonsmooth behavior is in part due to the uneven terrain. The dynamic inclusion of the new feature points is summarized in Table 4. Figure 15 shows the feature points detected in the 2^{nd}, 8^{th}, 13^{th}, 19^{th}, 25^{th} and 30^{th} frames and the points added to the tracking list.

Table 4: The number of feature points in the tracking list for the Martain Marietta R3 Sequence

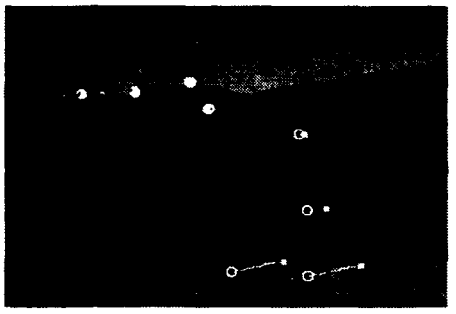
frame number	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
# of points in the list	0	9	12	18	21	20	23	22	25	25	27	27	30	30	28
# of points extracted	9	15	17	18	15	15	13	14	10	17	11	15	17	10	12
# of new points	9	4	8	3	0	3	0	3	1	4	0	3	2	0	2
frame number	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30
# of points in the list	27	29	30	31	32	33	33	33	37	38	37	37	37	39	37
# of points extracted	18	19	18	11	15	17	15	17	19	19	20	15	16	10	17
# of new points	2	1	4	1	2	2	1	4	1	2	3	3	5	0	3

4 Conclusions

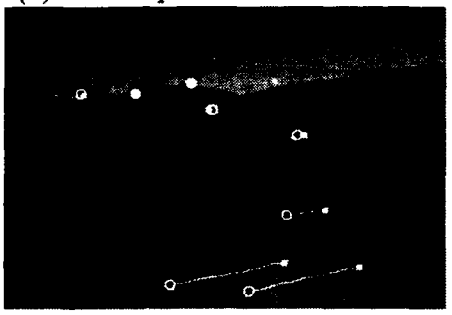
An algorithm for tracking a dynamic set of feature points to subpixel accuracy over a sequence of images has been presented. In particular, a simple 2-D kinematic motion model is employed to describe feature point trajectories, instead of a more complicated 3-D model. To account for deviations from the 2-D motion model, the PDAF is used to provide initial estimates of the in-frame motion parameters. The application of local image registration, taking into account the varying depth of the scene and compensating for the motion between two consecutive frames, reduces the search area and matching errors. In addition, the inclusion of new feature points makes the



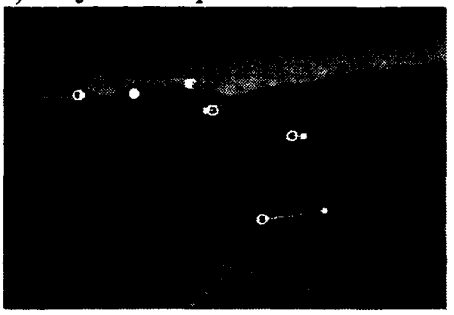
(a) Feature points in the first frame



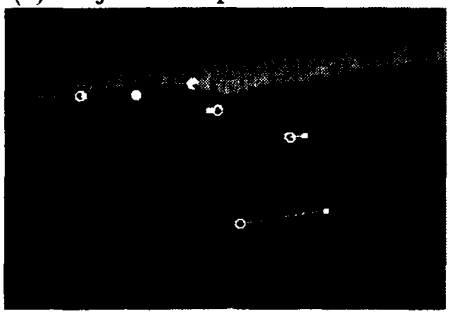
(b) Trajectories up to the seventh frame



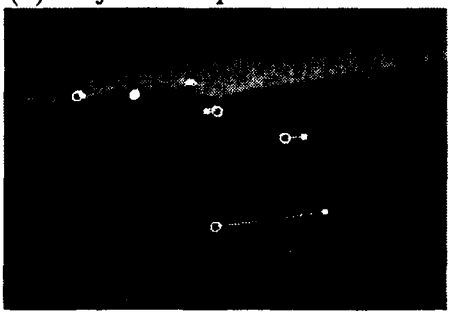
(c) Trajectories up to the 13^{th} frame



(d) Trajectories up to the 19^{th} frame



(e) Trajectories up to the 25^{th} frame



(f) Trajectories up to the 30^{th} frame

Figure 13: Trajectories for the Martin Marietta R3 Sequence

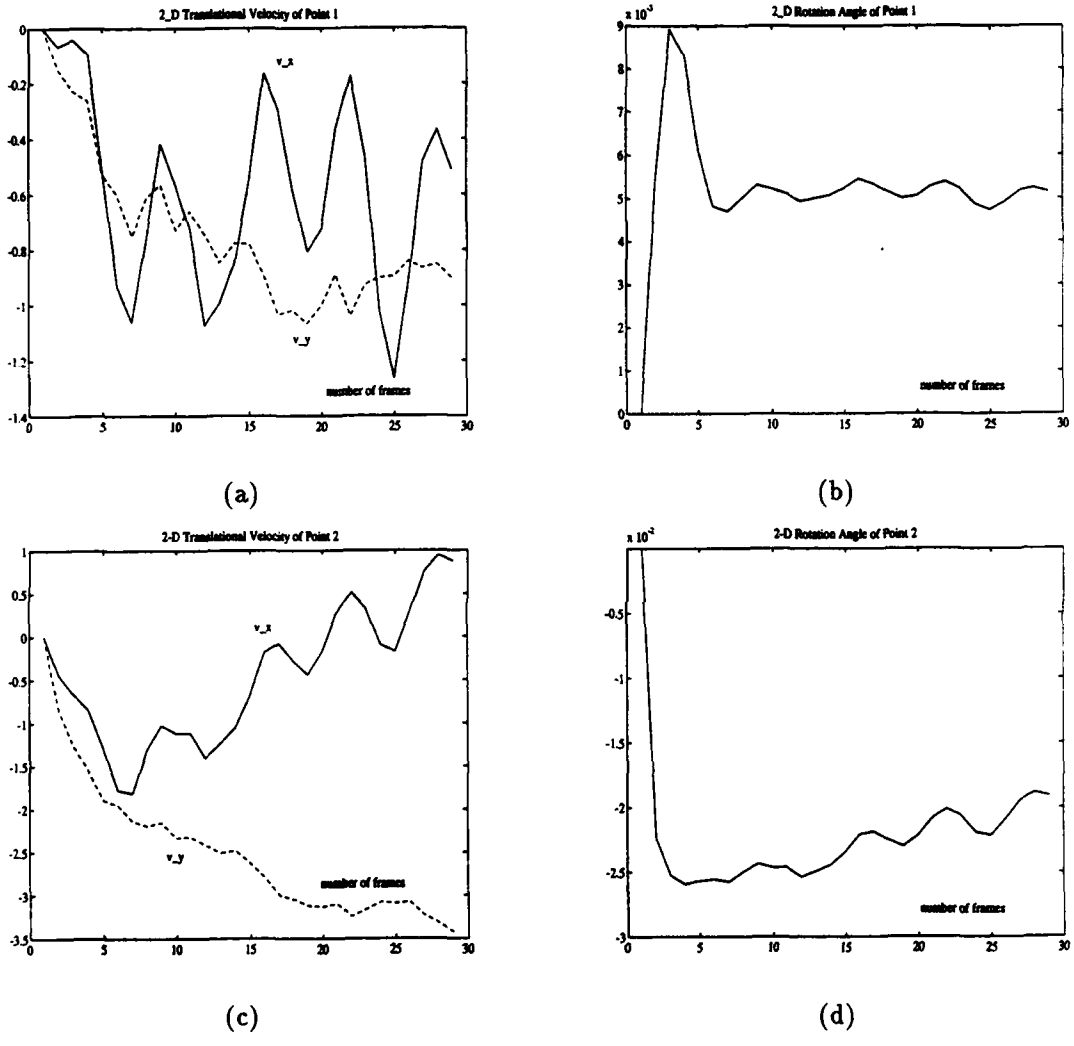
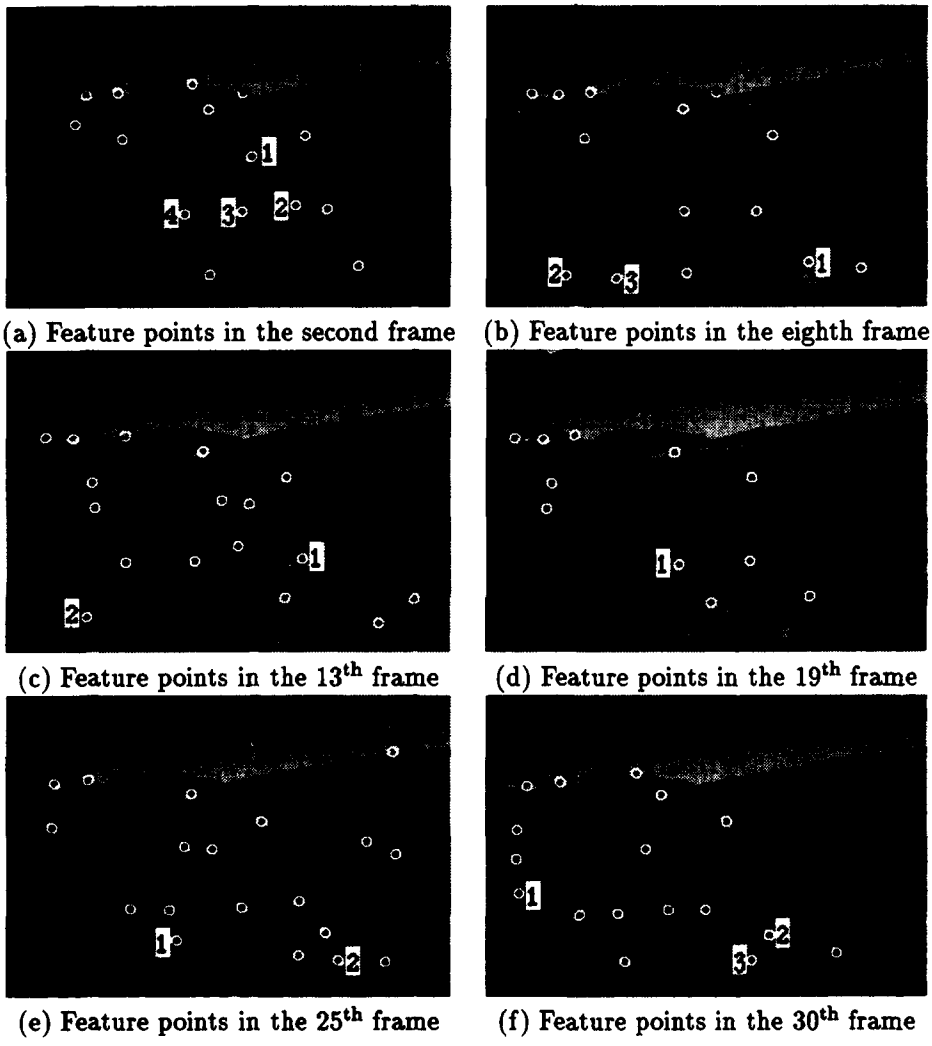


Figure 14: Motion parameters for the Martin Marietta R3 Sequence computed by the Kalman Filter



(a) Feature points in the second frame (b) Feature points in the eighth frame

(c) Feature points in the 13th frame (d) Feature points in the 19th frame

(e) Feature points in the 25th frame (f) Feature points in the 30th frame

Figure 15: Automatically detected new feature points in the Martin Marietta R3 Sequence

algorithm useful for the estimation of the pose and motion of the camera.

References

[1] J.K. Aggarwal, "Motion and Time-Varying Imagery—An Overview," in *Proc. IEEE Workshop on Motion: Representation and Analysis*, Kiawah Island, SC, pp. 1–6, May 1986.

[2] J.K. Aggarwal and A. Mitiche, "Structure and Motion from Images: Fact and Fiction," in *Proc. Third Workshop on Computer Vision: Representation and Control*, Bellaire, MI, pp. 127–128, Oct. 1985.

[3] Y. Bar-Shalom, "Tracking Methods in a Multitarget Environment," *IEEE Trans. Automatic Control*, Vol. AC-23, pp. 618–626, Aug. 1978.

[4] Y. Bar-Shalom and T.E. Fortmann, *Tracking and Data Association*, San Diego, CA: Academic Press, 1988.

[5] S.D. Blostein and T.S. Huang, "Detecting Small, Moving Objects in Image Sequences Using Sequential Hypothesis Testing," *IEEE Trans. Signal Processing*, Vol. 39, pp. 1611–1629, July 1991.

[6] T.J. Broida and R. Chellappa, "Estimating the Kinematics and Structure of a Rigid Object from a Sequence of Monocular Images," *IEEE Trans. Patt. Anal. Mach. Intell.*, Vol. PAMI-13, pp. 497–513, June 1991.

[7] Y.L. Chang and J.K. Aggarwal, "3D Structure Reconstruction From an Ego Motion Sequence Using Statistical Estimation and Detection Theory," in *Proc. IEEE Workshop on Visual Motion*, Princeton, NJ, Oct. 1991.

[8] I.J. Cox, "A Review of Statistical Data Association Techniques for Motion Correspondence," *International Journal of Computer Vision*, Vol. 10, pp. 53–66, Aug. 1993.

[9] N. Cui, J. Weng, and P. Cohen, "Extended Structure and Motion Analysis from Monocular Image Sequences," in *Proc. Third International Conference on Computer Vision*, Osaka, Japan, pp. 222–229, Dec. 1990.

[10] R.O. Duda and P.E. Hart, *Pattern Classification and Scene Analysis*, New York: Wiley, 1973.

[11] J.Q. Fang and T.S. Huang, "Some Experiments on Estimating the 3-D Motion Parameters of a Rigid Body From Two Consecutive Image Frames," *IEEE Trans. Patt. Anal. Mach. Intell.*, Vol. PAMI-6, pp. 545–554, Sept. 1984.

[12] T.S. Huang, ed., *Image Sequence Analysis*, Berlin/Heidelberg: Springer-Verlag, 1981.

[13] T.S. Huang *et al.*, "Motion Detection and Estimation from Stereo Image Sequences: Some Preliminary Experimental Results," in *Proc. IEEE Workshop on Motion: Representation and Analysis*, Kiawah Island, SC, pp. 45–46, May 1986.

[14] B.S. Manjunath, R. Chellappa, and C.V. Malsburg, "A Feature Based Approach to Face Recognition," in *Proc. IEEE Conf. on Computer Vision and Pattern Recognition*, Champaign, IL, pp. 373–378, June 1992.

[15] I.K. Sethi and R. Jain, "Finding Trajectories of Feature Points in a Monocular Image Sequence," *IEEE Trans. Patt. Anal. Mach. Intell.*, Vol. PAMI-9, pp. 56–73, Jan. 1987.

[16] J. Weng, N. Ahuja, and T.S. Huang, "Matching Two Perspective Views," *IEEE Trans. Patt. Anal. Mach. Intell.*, Vol. PAMI-14, pp. 806–825, Aug. 1992.

[17] J. Weng, T.S. Huang, and N. Ahuja, "3-D Motion Estimation, Understanding, and Prediction from Noisy Image Sequences," *IEEE Trans. Patt. Anal. Mach. Intell.*, Vol. PAMI-9, pp. 370–389, May 1987.

[18] T.H. Wu and R. Chellappa, "3-D Recovery of Structural and Kinematic Parameters from a Long Sequence of Noisy Images," in *Proc. ARPA Image Understanding Workshop*, Washington, DC, pp. 641–651, Apr. 1993. Accepted for publication, *International Journal of Computer Vision.*

[19] G.S. Young and R. Chellappa, "3-D Motion Estimation Using a Sequence of Noisy Stereo Images: Models, Estimation, and Uniqueness Results," *IEEE Trans. Patt. Anal. Mach. Intell.*, Vol. PAMI-12, pp. 735–759, Aug. 1990.

[20] Z. Zhang and O.D. Faugeras, "Three-Dimensional Motion Computation and Object Segmentation in a Long Sequence of Stereo Frames," *International Journal of Computer Vision*, Vol. 7, pp. 211–241, Aug. 1992.

[21] Q. Zheng and R. Chellappa, "Automatic Feature Point Extraction and Tracking in Image Sequences for Arbitrary Camera Motion," Tech. Rep. CAR-TR-628, Center for Automation Research, Univ. of Maryland, College Park, MD, 1992. Accepted for publication, *International Journal of Computer Vision.*

Appendix A

We describe the PDAF in this appendix. Assuming that the trajectory of a feature point over the image sequence has been established up to the k^{th} image, the time-varying behavior of the corresponding state vector between t_k and t_{k+1} as well as the relationship between the measurements and the state vector are the same as (2) and (4), i.e.

$$\begin{aligned} \underline{x}(k+1) &= \underline{f}[\underline{x}(k)] + \underline{w}(k+1) \\ \underline{z}(k+1) &= H\underline{x}(k+1) + \underline{n}(k+1) \end{aligned}$$

Since the plant equation is nonlinear, in order to linearize the nonlinear function $\underline{f}$ using the first order Taylor series expansion, the following matrix is defined:

$$F = \frac{\partial \underline{f}}{\partial \underline{x}}$$

Then at t_{k+1}, before taking into account any measurement, the PDAF [4] first propagates the state vector and the covariance matrix from t_k to t_{k+1} and predicts the location of the corresponding point, $\hat{\underline{z}}(k+1|k)$, by [4]

$$\begin{aligned} \hat{\underline{x}}(k+1|k) &= F[\hat{\underline{x}}(k|k)]\hat{\underline{x}}(k|k) \\ \hat{P}(k+1|k) &= F[\hat{\underline{x}}(k|k)]\hat{P}(k|k)F[\hat{\underline{x}}(k|k)]^T + Q(k+1) \\ \hat{\underline{z}}(k+1|k) &= H\hat{\underline{x}}(k|k) \end{aligned} \tag{34}$$

Subsequently, in order to incorporate the information contained in the $(k+1)^{\text{st}}$ image, a validation gate constructed based on the Mahalanobis distance in (7) is applied. Only the extracted points with distance less than a threshold, say γ, are considered as the possible corresponding point for the feature point. Without loss of generality, assume that there are m_{k+1} points inside the validation gate. The PDAF is ready to update the state vector and the covariance matrix using the past and present information.

Since there is an ambiguity in deciding which point among the m_{k+1} points corresponds to the feature point, the a posteriori probability of each point being correct given the past information is evaluated. In other words, the association probability for the j^{th} point, $\underline{z}_j(k+1)$, is defined as

$$\beta_j(k+1) = Pr[\theta_j(k+1)|Z^i(k+1)] \tag{35}$$

where $Z^i(k+1)$ is the collection of all possible measurements at each time instant and

$$\theta_j(k+1) \equiv \{\underline{z}_j(k+1) \text{ is the corresponding point}\}$$

In addition, the probability that none of the m_{k+1} points is correct is considered and denoted by $\beta_0(k+1)$. As above, if each point is assumed to be normally distributed about $\hat{\underline{z}}(k+1|k)$ with the corresponding innovation vector represented by $\underline{v}_j(k+1)$, the association probabilities are shown to be the following [4]:

$$\begin{aligned} \beta_j(k+1) &= \frac{e_j}{b+\sum_{l=1}^{m_{k+1}} e_l} \quad j = 1, \ldots, m_{k+1} \\ \beta_0(k+1) &= \frac{b}{b+\sum_{l=1}^{m_{k+1}} e_l} \end{aligned} \tag{36}$$

where

$$\begin{aligned} e_j &= \exp\left[-\tfrac{1}{2}\,\underline{v}_j(k+1)^T S(k+1)^{-1}\underline{v}_j(k+1)\right] \\ b &= \left(\frac{2}{\gamma}\right) m_{k+1}\frac{1-P_d P_g}{P_d} \end{aligned} \tag{37}$$

In (37), P_g and P_d represent the a priori probabilities that an extracted point falls into the validation gate and that the corresponding point is detected by the feature extraction algorithm respectively. These prior probabilities are set to be the same for each feature point being tracked.

After the association probabilities are obtained, the state vector as well as the covariance matrix are updated by combining the information contained in the m_{k+1} points with the predicted estimates as follows:

$$\begin{aligned} K(k+1) &= \hat{P}(k+1|k)H^T[H\hat{P}(k+1|k)H^T + R(k+1)]^{-1} \\ \hat{\underline{x}}(k+1|k+1) &= \hat{\underline{x}}(k+1|k) + K(k+1)\underline{v}(k+1) \\ \hat{P}(k+1|k+1) &= \beta_0(k+1)\hat{P}(k+1|k) + [1-\beta_0(k+1)]P^c(k+1|k+1) + \tilde{P}(k+1) \end{aligned} \tag{38}$$

where

$$\begin{aligned} \underline{v}(k+1) &= \textstyle\sum_{j=1}^{m_{k+1}} \beta_j(k+1)\underline{v}_j(k+1) \\ \tilde{P}(k+1) &= K(k+1)[\textstyle\sum_{j=1}^{m_{k+1}} \beta_j(k+1)\underline{v}_j(k+1)\underline{v}_j(k+1)^T - \underline{v}(k+1) \\ &\quad \underline{v}(k+1)^T]K(k+1)^T \\ P^c(k+1|k+1) &= [I - K(k+1)H]\hat{P}(k+1|k) \end{aligned} \tag{39}$$

This completes one cycle of the PDAF. As seen in (38), the uncertainties in the measurements result in the m_{k+1} extracted points being combined with different weights to correct the predicted state vector in (34), and the uncertainties in the state estimates are increased as shown in (38).

It is worth noting that in the derivation of the PDAF, all measurements falling within a validation gate are considered equally likely to be the correct measurements. However, if additional information is available so that the ambiguity can be resolved and one of the measurements, say z_j, is identified as the correct measurement, its corresponding association probability, $\beta_j(k+1)$, should be set to 1. This leads to the well known EKF.

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

1. AGENCY USE ONLY (Leave blank)	2. REPORT DATE	3. REPORT TYPE AND DATES COVERED
	June 1994	Technical Report

4. TITLE AND SUBTITLE
Tracking a Dynamic Set of Feature Points

5. FUNDING NUMBERS
DAAH-0493G0419

6. AUTHOR(S)
Yi-Sheng Yao and Rama Chellappa

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)
Computer Vision Laboratory
Center for Automation Research
University of Maryland
College Park, MD 20742-3275

8. PERFORMING ORGANIZATION REPORT NUMBER

9. SPONSORING / MONITORING AGENCY NAME(S) AND ADDRESS(ES)
U.S. Army Research Office
P.O. Box 12211
Research Triangle Park, NC 27709-2211

10. SPONSORING / MONITORING AGENCY REPORT NUMBER

11. SUPPLEMENTARY NOTES
The views, opinions and/or findings contained in this report are those of the author(s) and should not be construed as an official Department of the Army position, policy, or decision, unless so designated by other documentation.

12a. DISTRIBUTION / AVAILABILITY STATEMENT
Approved for public release; distribution unlimited.

12b. DISTRIBUTION CODE

13. ABSTRACT *(Maximum 200 words)*

This paper presents a model-based algorithm for tracking feature points over a long sequence of monocular noisy images with the ability to include new feature points detected in successive frames. The trajectory for each feature point is modeled by a simple kinematic motion model. A Probabilistic Data Association Filter is first designed to estimate the motion between two consecutive frames. A matching algorithm then identifies the corresponding point to subpixel accuracy and an Extended Kalman Filter (EKF) is employed to continually track the feature point. An efficient way to dynamically include new feature points from successive frames into a tracking list is also addressed. Tracking results for several image sequences are given.

14. SUBJECT TERMS	15. NUMBER OF PAGES
Feature point tracking, image sequence analysis, motion analysis	38
	16. PRICE CODE

17. SECURITY CLASSIFICATION OF REPORT	18. SECURITY CLASSIFICATION OF THIS PAGE	19. SECURITY CLASSIFICATION OF ABSTRACT	20. LIMITATION OF ABSTRACT
UNCLASSIFIED	UNCLASSIFIED	UNCLASSIFIED	UL